Research Article

Adaptive Reward Allocation for Participatory Sensing

Martin Connolly¹, **Ivana Dusparic²**, **Georgios Iosifidis²**, and **Melanie Bouroche²**

¹Department of Information Systems, Cork Institute of Technology, Ireland

²School of Computer Science & Statistics, Trinity College Dublin, Ireland

Correspondence should be addressed to Melanie Bouroche; melanie.bouroche@scss.tcd.ie

Received 30 April 2018; Accepted 25 July 2018; Published 7 August 2018

Academic Editor: Giovanni Stea

Copyright © 2018 Martin Connolly et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Participatory sensing is a paradigm through which mobile device users (or participants) collect and share data about their environments. The data captured by participants is typically submitted to an intermediary (the service provider) who will build a service based upon this data. For a participatory sensing system to attract the data submissions it requires, its users often need to be incentivized. However, as an environment is constantly changing (for example, an accident causing a buildup of traffic and elevated pollution levels), the value of a given data item to the service provider is likely to change significantly over time, and therefore an incentivization scheme must be able to adapt the rewards it offers in real-time to match the environmental conditions and current participation rates, thereby optimizing the consumption of the service provider's budget. This paper presents adaptive reward allocation (ARA), which uses the Lyapunov Optimization method to provide adaptive reward allocation that optimizes the consumption of the service provider's budget. ARA is evaluated using a simulated participatory sensing environment with experimental results showing that the rewards offered to participants are adjusted so as to ensure that the data captured matches the dynamic changes occurring in the sensing environment and takes the response rate into account while also seeking to optimize budget consumption.

1. Introduction

Participatory sensing is a form of crowdsourcing whereby individuals and communities submit scalar and/or multimedia data from mobile devices such as personal smart phones. The submitted data can be GPS coordinates revealing location or trajectory, a sensed data measurement, or multimedia content such as photos, sound clips, or video. The wide range of data that can be captured by participatory sensing is reflected in the diversity of its applications including, among others, smart cities [1], air pollution exposure [2], and health [3].

The key to the success of a participatory sensing application is attracting a critical mass of relevant data. However, while participants may be willing to make data submissions, the majority expect some form of reward in return [4]. These rewards could be monetary [4] or credit tokens that can then be used to claim a reward [5].

The issue of incentivization has direct implications for the quality of a service provider's dataset. Users who are paid

for assigned tasks complete them significantly more quickly than volunteer users [6]. In the case of participatory sensing and other similar data sharing environments, it has been found that proper incentive allocation improves data quality ([7, 8]). However, incentivization schemes for participatory sensing face a number of challenges to ensure that the service provider's dataset is relevant and timely. In particular, the conditions in a participatory sensing environment can suddenly change; for example, a bridge connecting two areas of a city being closed due to high winds would result in a buildup of traffic. As a result of such sudden changes, the utility value of a particular type of data submission to the service provider can also change significantly. Moreover, as participation rates will vary over time, the service provider needs the ability to adapt the level of reward it offers to match the current response rate. At the same time, a service provider will have a finite budget and will want to optimize its consumption of this budget. This is of benefit not only to the service provider but also to those participants who want to consume the data and will therefore want it to be as

relevant and timely as possible. Finally, any incentivization scheme should not have any negative impact upon other areas of concern in participatory sensing. For example, while there are reputation based incentivization schemes in the state of the art (for example, [9, 10]) that address the issue of data quality, this is typically at the expense of participant privacy.

This paper presents adaptive reward allocation (ARA), an incentivization scheme that is designed for participatory sensing environments. The approach is designed in such a way that it can be integrated for any type of participatory sensing system application without impinging upon other issues of concern such as privacy. ARA addresses the need for ongoing reward computation and allocation in a way that adapts to the detection of sudden changes in the dynamic fast moving environment in which participatory sensing applications operate. In particular, the reward level is adapted on the basis of the response rate to an offer previously made by the service provider and on the basis of the utility of the data.

2. Related Work

The majority of incentivization schemes in the state of the art for participatory sensing use microeconomics, statistics, or a combination of both for reward computation. Section 2.1 describes the economic approaches used for incentivization while Section 2.2 explores the options available from the field of statistics.

2.1. Economic Approaches to Incentivization. Several economic approaches in the state of the art use auctions ([8, 11–14]). There are many merits to the use of auctions including the diverse set of approaches that are available and its well-established use in incentive computation. However, auctions are vulnerable to collusion attacks [15]. In a participatory sensing environment, this means that colluding participants could consume a disproportionate amount of the service provider's budget, thus diminishing the quality of the overall dataset. Auctions also entail a high level of overhead [16] as, in a participatory sensing environment, the service provider will typically need to gather all bids before deciding which participants to select. Moreover, auction-based schemes violate privacy as, even if pseudonyms are used, the service provider can monitor participants' bid activity.

In addition, some of the auction-based incentivization approaches in the state of the art have attributes which further limit their efficacy. For example, the Vickrey-Clarke-Groves (VCG) auction policy [11], a type of sealed bid auction, assumes that participants do not consider future benefits when making a bid. The potential for this assumption to be violated is acknowledged by the authors themselves who point out that the service provider's budget would not be optimally consumed as a result. Similarly, the Cooperative Incentive Mechanism [13] exhibits some key limitations as its primary objective is to ensure that as many participants as possible are rewarded. While the authors justify this goal in terms of motivating participation, it does not take the optimization of budget consumption or the quality of the data submitted into account.

Other approaches in the state of the art use principles of microeconomics to construct their incentivization schemes. For example, SenseUtil [17] uses the concepts of supply and demand and marginal utility to determine the value of sensed data but does not attempt to optimize rewards to determine a level at which data submissions will be made below that value. Moreover, neither the independent utility metric (which is used to evaluate the uniqueness of the sensed data) nor the history-based utility metric (which evaluates the similarity of sensed data to other data submissions) seeks to capture sudden changes in the participatory sensing environment as budget consumption and adaptiveness to response rate are not considered.

Elementary supply and demand is also used to determine the grades of data being sought [18] where the incentivization approach is alluded to as a market mechanism, albeit without any further details. Similarly, SEQTGREEDY also applies microeconomics in its incentivization approach. In this case, the concept of marginal utility is used to maximize the service provider's marginal gain [19]. However, the means by which the reward level can be adapted to capture sudden changes in the participatory sensing environment and reflect current participation rates is not addressed by either approach.

2.2. Statistical Approaches to Incentivization. The statistical methods used for incentive computation in the state of the art for participatory sensing are principally based on probability ([20, 21]), optimization ([22, 23]), and stochastics ([11, 19, 21, 24, 25]). Many of these approaches use a combination of these methods for their incentivization mechanism, often in conjunction with microeconomic techniques.

There are a number of statistical-based incentivization approaches in the state of the art that seek to optimize budget consumption. For example, simulated annealing, a probabilistic optimization method, is used by EPPI [23] to minimize the levels of reward given to participants. The constrained budget of service providers is also taken into account. However, the approach does not consider the current participation rate and the dynamic changes that may occur in the participatory sensing environment.

Other approaches do consider the quality of the data submitted. For example, the Bayesian Truth Serum incentive scheme [20] evaluates the data submitted using a probabilistic scoring system. There is also a Gur Game based approach [22] (a mathematical modelling of what is termed reward and punishment) in the state of the art that takes the quality of data into account. However, these approaches neither address the optimal consumption of the service provider's budget nor adapt the reward level to reflect environmental changes or participation rates.

The stochastic-based "Backpressure Meets Taxes" (BMT) mechanism [24], which uses Lyapunov Optimization in conjunction with Mechanism Design (the former is used for sensing rate control and routing, not reward computation), also does not consider how to optimize the level of reward to offer to participants and hence does not address the optimizing of budget consumption. Rather, BMT seeks to maximize what it terms the gross profit of participants. In contrast, the Markov Model based incentivization

negotiation mechanism outlined in the state of the art [21] estimates the probability of data collection using this model and also applies the economic concept of supply and demand to create a budget optimization policy that takes the quality of the data submitted into account. However, while this approach does take budget constraints into account, its focus is on reward allocation fairness and the geographical distribution of the budget across the subregions of the sensed environment rather than the data that is of most relevance to a service provider at a particular point in time.

SEQTGREEDY [19] also uses stochastics in conjunction with microeconomic concepts. In this case, the Stochastic Submodular Maximization method is used to enable privacy trade-offs in return for a reward. However, the assumption of diminishing returns inherent in the submodular functions used for this technique (i.e., incremental returns are lower over time) is not appropriate to adaptive reward allocation as, in the case of participatory sensing, this implies that the service provider could pay a higher level of reward over time.

The STOC-PISCES algorithm [25] applies binary search and the Multi-Armed Bandit (MAB) Framework, a probabilistic method of resource allocation, to the stochastic (i.e., uncertain) setting of reward minimization. STOC-PISCES addresses the problem of determining the optimal reward level and also takes the utility of data (characterized as the demand for a particular type of data submission) into account when computing rewards. Moreover, the participation rate is taken into account as the underlying PISCES framework defines a reward for data submission at the beginning of a number of what it terms a trial and then adapts the reward for subsequent trials until the desired number of data submissions is obtained. However, while the approach sets a minimum and maximum range for the reward to offer, it does not seek to optimize budget consumption. As illustrated elsewhere in the state of the art, the goal of budget optimization would require substantial modification of the underlying MAB algorithm [23].

To conclude, incentivization has been considered in the state of the art with SenseUtil and the STOC-PISCES approaches in particular addressing the need to consider participation rates and data utility. However, there is a need for an adaptive reward allocation scheme that not only takes both participation rates and data utility into account but also seeks to optimize consumption of a finite budget, i.e., an approach that seeks to optimize the trade-off between the number of responses received and the budget consumption.

3. System Model and Assumptions

The participatory system model assumed for ARA has two actors: the service provider and the participant. A typical service provider wants to receive timely and relevant scalar and/or multimedia data pertaining to an environment. The service provider will publish the data it is interested in as a task to which participants can elect to respond in return for a reward. A task can consist of single or multiple types of sensed scalar or multimedia data. Depending on the nature of the service provider, data submissions can then be consumed by other users (for example, current pollution levels in a city)

or used for data analytics purposes (for example, to build a climate model).

The participatory sensing environment used for ARA is modelled as a service provider issuing offers to participants with offers consisting of the data being sought and the reward given for data matching that is requested by the offer. If the reward is greater than or equal to the minimum reward expected by a participant, that participant will then decide whether to make a data submission in response to this offer. It is assumed that participants are rational; i.e., the higher the reward offered for a particular type of data, the larger the number of responses (assuming other factors such as privacy perceptions remain constant). The participants are also assumed to incur costs (for example, battery consumption, consumption of network provider's user data allocation) when making data submissions.

It is assumed that the service provider's budget is finite. This budget will either be a monetary one or consist of tangible rewards (for example, Wi-Fi access). A participant only receives a reward on full completion of a task with rewards only being allocated until the service provider has received its desired number of responses.

The fundamental problem being addressed by ARA is a time average cost minimization one as the service provider is seeking to set the offered reward and corresponding budget consumption at the minimum level that will attract an acceptable level of relevant responses from participants. To model the problem, it is assumed that the service provider operates in discrete time over slots $t \in 1, 2, \dots$ with the reward level being reviewed at the start of each time slot. The service provider can issue one or more offers seeking data submissions in a time slot, t . Offers can be categorised by different levels of granularity of the service provider's choosing, for example, the level of privacy to be ceded or location accuracy.

4. Adaptive Reward Allocation (ARA)

This section describes the adaptive reward allocation (ARA) scheme for computing rewards in participatory sensing environments. Section 4.1 discusses why Lyapunov Optimization is used as the foundation for ARA's incentivization scheme while Section 4.2 describes how supply curves are used to estimate the number of responses. Section 4.3 outlines how the participatory sensing environment is modelled in order to determine the number of responses that will be made for different levels of reward. Section 4.4 then formulates the budget optimization problem to be addressed. Sections 4.5 and 4.6 describe the offline and online budget consumption problems, respectively, while the design of the online algorithm is outlined in Section 4.7. Finally, the incorporation of data utility into the algorithm is discussed in Section 4.8.

4.1. Lyapunov Optimization. The method used by ARA for ongoing reward allocation is based upon Lyapunov Optimization. Lyapunov Optimization is a method that is particularly suitable for the controlling of dynamic systems. It is used for the computation of incentives and pricing in communication networks and has been previously used for

incentive design for participatory sensing [26] (though not reward computation). It can be used to minimize dynamic costs [27] and is suitable for rapid changes over time in the environment in which it is applied [28]. These attributes are directly relevant given the desire by service providers that budget consumption be optimized.

Lyapunov Optimization is particularly appropriate for ARA as the approach seeks to dynamically adapt rewards so as to respond to sudden and rapid changes in an environment with the nature, accuracy, quality, and level of detail of the data varying depending on the circumstances. Furthermore, the fact that a Lyapunov Optimization solution at any one time affects the constraint to be applied the next time the optimization is carried out is important for ARA as the service provider's budget is being consumed with each optimization solution that results in accepted offers. Finally, the use of Lyapunov Optimization does not require future knowledge of the rate of response to offers made by the service provider. This is crucial for ARA's reward model.

As Lyapunov Optimization is principally used for resource allocation problems in domains such as computer networking [29], its use must be modified for the problem ARA is seeking to address. This is principally because there are a number of differentiating attributes of an economic market in participatory sensing. In particular, the data being sought by the service provider (equivalent to the product in other economic markets) can potentially change suddenly and its value to the service provider will change depending on that party's needs at a particular point in time. While demand may change in other price optimization scenarios such as wind power or cloud infrastructure rental, the product does not. In participatory sensing, the "product" (type of data sought) not only changes over time but is time sensitive and needs to match the information sought by the service provider [30]. It is thus an appropriate candidate for a market-based model.

4.2. Estimating the Number of Responses. The reward included in the offer published by the service provider is a key factor in determining the number of responses, $N_O(t)$, for each offer O . It is therefore assumed that $N_O(t)$ is a function f of the offered reward denoted by $R_O(t)$ (all participants are offered the same reward):

$$N_O(t) = f(R_O(t)) \quad (1)$$

To estimate $N_O(t)$, ARA requires a dataset that it can use to compute the appropriate value for $R_O(t)$. In microeconomic terms, this is the *reservation price* at which the participant is willing to "sell" data. While the reservation price is typically computed by methods such as the Conjoint Analysis [31] and Contingent Valuation methods [29], these methods are dependent upon surveying potential customers (or participants in this case) which is not a practical option for meeting the requirement for adaptive reward allocation in a participatory sensing environment. Instead, ARA builds up a picture of participants' willingness to accept offers at particular rates from supply curves that use previous data submissions from the service provider's existing dataset. Previous data submissions thus act as a substitute for a survey

to present an ongoing evolving picture of the willingness to accept offers at particular levels of reward.

As the level of reward set by the service provider is a key determinant of the number of data submissions it obtains in response to an offer, the above function can be modelled using the microeconomic concept of a supply curve. The formal definition of a supply curve is a graphic representation of the relationship between product price and the quantity of the product that a seller is willing and able to supply. In terms of the ARA model, a number of supply curves are used to estimate the relationship between the reward offered and the number of responses different categories of offers attract from participants. These supply curves evolve over time as more offers are made by the service provider and more responses to offers are received. The relationship between the number of responses and the reward level thus serves as ever evolving training data (a set of data used to discover relationships) to enable the service provider to more accurately estimate the reward that will generate its desired number of responses.

Each supply curve is modelled using regression analysis to predict the willingness of participants to accept offers at different reward levels. Typically, both demand and supply are modelled as a function of price and cost, respectively, using linear regression in the field of Econometrics [32] (see also, for example, [33]). However, to facilitate the incorporation of other predictors that will not necessarily have a linear relationship (for example, the effort involved in capturing the data), a nonlinear multiple regression method is used to predict the number of responses, N_{predict} . Specifically, a rolling window time series regression model is used to construct the prediction model so that only the most recent data is taken into account in the simulation. The size of the rolling window used can be altered depending on the circumstances in the participating sensing environment without impacting the algorithm. Indeed, any form of predictive modelling technique can be used to update the supply curves, thus allowing the service provider to evaluate which is the best predictive model to use [34].

As noted in Section 3, the participant will incur costs when submitting data resulting in different willingness to make data submissions. These costs can be considered as random effects that are summarized as a cost parameter C , which is random, i.i.d (independent and identically distributed) and varies between time slots. When the cost is high (for example, the smartphone is required for the user's own needs; the battery is low), the user needs a higher reward to participate. When C is low (for example, the device is idle; the user has time to complete the task) then even a low reward might be enough. While the service provider does not have access to each individual participant's circumstances during a particular time slot, it can nevertheless estimate C in terms of, for example, battery consumption, data transmission costs, and latency, i.e., the time taken to accept a task, carry out a task, make a data submission, and receive the reward for the completion of the task.

The number of current active participants P in each time slot t is another parameter of interest when predicting the number of responses. For example, when there are many active participants, a small reward that can motivate only 10%

of these users might be enough in order to ensure the required number of responses. On the other hand, a higher per user reward is necessary for a participatory system with less active participants.

Therefore, N_{predict} can be defined in terms of the rewards offered R , the cost of carrying out the task, $-C$, and the ratio of the number of responses sought to the current number of participants, P_{ratio} . Using X to denote this set of predictors as a vector and β to denote a vector of parameter coefficients, N_{predict} can be expressed as follows:

$$N_{\text{predict}} = f(X, \beta) + \varepsilon \quad (2)$$

where ε is an error term.

Equation (2) can be expanded to incorporate R , $-C$, and P_{ratio} . In addition, while the problem is nonlinear, it can be expressed in epigraph form as follows:

$$N_{\text{predict}} = \beta_0 R - \beta_1 C + \beta_2 P_{\text{ratio}} \quad (3)$$

where

β_0 is the regression coefficient for R .

β_1 is the regression coefficient for $-C$.

β_2 is the regression coefficient for P_{ratio} .

Equation (3) can be extended by the service provider to incorporate other coefficients if there are other factors that determine the number of responses, for example, the level of privacy to be ceded. In addition, the service provider can remove what it deems to be irrelevant predictors without impacting the underlying reward model. For example, a service provider who is only seeking scalar data such as temperature might consider the task cost to be broadly similar between time slots.

It should be noted that if the number of responses is greater than N_{desired} , it will only be desirable from the service provider's perspective to reward some of the responses to an offer. Moreover, while it may be possible to attract $N_{\text{desired}}(t)$, this might necessitate a reward level that is not consistent with optimal consumption of the service provider's budget. Hence, while the supply curves can be used to determine reward levels, the trade-off between achieving $N_{\text{desired}}(t)$ and budget consumption must be addressed. It is thus necessary to model this trade-off for the participatory sensing environment.

4.3. Modelling the Environment for Reward Determination.

The relationship assumed by (1) is used to build up a picture of the (estimated) number of participant responses to a particular reward. However, there will be a point at which increasing the reward will not lead to an increase in the number of responses even if parameters such as C remain unchanged. This is because the maximum number of responses is equal to the number of participants in the participatory sensing system $P(t)$ and varies over time as participants join and leave the system (either by formally

deregistering or ceasing to participate). Thus for every time slot t

$$0 \leq N_O(t) \leq P(t) \quad (4)$$

$R_P(t)$ denotes the reward level when the number of responses equals the number of participants, i.e., when demand equals supply in economic terms:

$$N_O(t) = P(t) \quad (5)$$

$P(t)$ is upper bounded by a constant P_{max} which corresponds to the number of participants potentially active on the system. This leads to the following constraint for every time slot t :

$$0 \leq P(t) \leq P_{\text{max}} \quad (6)$$

Using the supply curves, ARA can estimate the number of responses that should be received at different levels of rewards for different categories of data. For example, the service provider estimates that it will receive N_O number of responses when the reward level is set to R_O . Taking (5) and (6) into account, R_O should not exceed $R_P(t)$ as exceeding $R_P(t)$ will not increase the number of responses:

$$0 \leq R_O \leq R_P(t) \quad (7)$$

As the supply curves evolve over time, the process of updating each curve is undertaken at the beginning of each time slot when reviewing the reward level. The service provider uses the reward-response data it has observed over previous time periods and, accordingly, updates the supply curve for this time slot.

The problem ARA is seeking to address can thus be defined as follows:

Problem Definition. For a given number of responses in a time slot t that follows an i.i.d. process with mean cost C , and for a certain level of minimum participants that the system should recruit, design a dynamic algorithm that finds the optimal level of reward so as to satisfy the above constraints while minimizing the budget consumption of the service provider.

To achieve a trade-off between minimizing the number of offers forfeited due to too low a reward and optimizing budget consumption, the former is defined as a queue for a time slot t , $Z_{\text{forfeit}}(t)$ (Z is used to denote a virtual queue as this notation corresponds to that used in [35]). The number of forfeited responses is what is termed a "virtual queue". As the name implies, virtual queues do not exist in reality and are only implemented in software to facilitate the definition of the Lyapunov Optimization-based model [29].

$Z_{\text{forfeit}}(t)$ is computed in terms of the number of responses desired by the service provider, N_{desired} . Thus, in any time slot, t , $Z_{\text{forfeit}}(t)$ is the difference between the actual number of responses received $N_{\text{received}}(t)$ and the desired number of responses $N_{\text{desired}}(t)$:

$$Z_{\text{forfeit}}(t) = \begin{cases} 0 & N_{\text{received}}(t) \geq N_{\text{desired}}(t) \\ \min\{N_{\text{desired}}(t), P(t)\} - N_{\text{received}}(t) & N_{\text{received}}(t) < N_{\text{desired}}(t) \end{cases} \quad (8)$$

4.4. Modelling the Environment for Budget Optimization. As originally formulated, Lyapunov Optimization is used to minimize the backlog of a queue for the purposes of optimizing resource allocation [35]. In mathematical terms, the method is the sum of squares of the queue (multiplied by 1/2) arising from a resourcing problem:

$$L(t) = \left(\frac{1}{2}\right) \sum_i Z_i(t)^2 \quad (9)$$

Equation (9) measures the queue backlog for the system model, the queue being the number of forfeited responses as defined by (8).

The computation of $N_{\text{desired}}(t)$ is dependent upon the requirements of the service provider. In a fast changing environment, it could decide that its desired number of responses is determined by its needs at a particular time; i.e., for every time slot t , the desired number of responses is independent of previous timeslots:

$$N_{\text{desired}}(t) \perp N_{\text{desired}}(t-1) \quad (10)$$

In such a case, it is assumed that $N_{\text{desired}}(t)$ is i.i.d. over the time slots. Furthermore, unlike other scenarios typically modelled using Lyapunov Optimization (for example, [27]), $Z_{\text{forfeit}}(t)$ is, for every time slot t , independent of queue backlogs from previous timeslots:

$$Z_{\text{forfeit}}(t) \perp Z_{\text{forfeit}}(t-1) \quad (11)$$

Alternatively, the service provider may decide that if, for a previous timeslot $t-1$, $N_{\text{desired}}(t-1) < N_{\text{received}}(t-1)$, $N_{\text{desired}}(t)$ is determined by $N_{\text{desired}}(t-1)$; i.e., for every time slot t

$$\begin{aligned} N_{\text{desired}}(t) &= N_{\text{desired}}(t-1) - N_{\text{received}}(t-1) \\ \text{s.t. } N_{\text{desired}}(t-1) &> N_{\text{received}}(t-1) \end{aligned} \quad (12)$$

This then implies that the value of $Z_{\text{forfeit}}(t)$ is determined by $Z_{\text{forfeit}}(t-1)$; i.e.,

$$Z_{\text{forfeit}}(t) = f(Z_{\text{forfeit}}(t-1)) \quad (13)$$

While the underlying probability distribution and other statistical characteristics of $N_{\text{desired}}(t)$ are not known by the service provider and are not required for Lyapunov Optimization, it must be assumed that its maximum value is finite:

$$0 \leq N_{\text{desired}}(t) \leq N_{\text{desired,max}} \quad (14)$$

Moreover, a further assumption is that the number of received responses to offers is bounded by the number of potentially active participants in the system. Thus, the expected values (the long run average values) of $N_{\text{desired}}(t+1)$ and $P(t)$ adhere to the following rule:

$$E(N_{\text{desired}}(t+1)) \leq E(P(t)) \quad (15)$$

This inequality ensures that there is a reward allocation schedule that ensures the stability of $Z_{\text{forfeit}}(t)$. Using the

rate stability theorem [35], $\bar{Z}_{\text{forfeit}}$ is used to denote the time average queue backlog for the forfeited responses. The stability of the queue is equal by definition as follows:

$$\bar{Z}_{\text{forfeit}} \triangleq \lim_{t \rightarrow \infty} \frac{1}{t} \sum_{i=0}^{t-1} E\{Z_{\text{forfeit}_k}(t)\} < \infty \quad (16)$$

It is assumed that the reward is upper bounded by a constant R_{max} . This means that for all time slots t

$$0 \leq R(t) \leq R_{\text{max}} \quad (17)$$

In addition, the service provider can also set the maximum of proportion of the budget, $B_{\text{proportion,max}}$, that can be consumed for an offer in a given time slot:

$$0 \leq N(t) R(t) \leq B_{\text{proportion,max}} \quad (18)$$

4.5. Formulating the Offline Problem. Before modelling the budget consumption problem for ARA, it is necessary to establish benchmarks to evaluate the approach. This section formulates the problem of reward allocation as two offline problems with complete future information and stochastic information, respectively, as benchmarks. These benchmark cases assume information symmetry; i.e., the service provider knows the response rate for a particular reward in the case of full information and knows the budget consumption under different scenarios in the case of stochastic future information.

4.5.1. Complete Future Information. With complete future information, the service provider can determine the response rate jointly in all time slots to minimize budget consumption. To formulate the offline budget consumption problem, T is defined as the set of all time slots $1..t_n$ during the sensing period where t_n represents the final time slot. As no linear relationship is assumed between the number of responses and the reward offered, the problem is a nonlinear convex optimization problem and can be formulated as follows for an individual timeslot t :

$$\begin{aligned} \min_{t \in T} \quad & \{N_1(t) R_1(t) \dots N_{\text{max}}(t) R_{\text{max}}(t)\} \\ \text{s.t.} \quad & R(t) \leq R_{\text{max}} \\ & N(t) \geq N_{\text{desired}} \\ & N(t) R(t) \leq B_{\text{remain}} \end{aligned} \quad (19)$$

where R_{max} is the maximum reward.

N_{max} is the number of responses received for R_{max} .

B_{remain} is the remaining budget.

The problem of minimizing the budget consumption over the entire set of time slots T is subject to the same constraints and is formulated as follows:

$$\frac{1}{T} \sum_{t \in T} N(t) R(t) \quad (20)$$

The offline reward allocation problem solved in (20) incorporates the explicit response rate of every time slot in advance.

There are a wide range of optimization methods that can be used to solve (20), for example, the first fit and best fit algorithms, nonlinear programming methods, mixed integer linear programming methods (by formulating the problem in linear epigraph form), or, by using the linear programming relaxation, the simplex method or KKT analysis [8].

The formulation and solving of (20) require complete knowledge of the future response rate in every time slot t , which is obviously impractical. For this reason, a model which only requires certain future information is defined.

4.5.2. Stochastic Future Information. This section proposes a benchmark based on stochastic future information where the response rate for each time slot follows the same probability space. With stochastic information only, the service provider cannot decide the reward for a timeslot in advance as it does not have complete future information. This case focuses on the expected budget consumption optimization based on stochastic information.

Θ defines the set of possible scenarios (or information realizations) that can occur when a service provider makes an offer at a particular reward level, R . $R(\theta)$ and $N(\theta)$, respectively, denote the reward level and the number of expected responses to that reward under a particular information realization θ . Budget consumption under θ is $N(\theta)R(\theta)$. Therefore, the expected budget optimization problem can be defined as follows:

$$\min \int_{\theta \in \Theta} N(\theta) R(\theta) \quad (21)$$

Like (20), (21) is an offline problem subject to the same constraints that in this case defines a contingency plan that specifies the budget consumption under each information realization θ . It is a nonlinear programming problem with an infinite number of variables as θ is continuous.

4.5.3. Analysing the Benchmarks. The next step is to analyse the gap between the minimum budget consumption with complete future information derived from (20) and the minimum budget consumption with stochastic future information derived from (21). These are denoted by B^o and B^* , respectively. As indicated in the state of the art [8], this can be expressed formally as follows.

Lemma 1. *If $T \rightarrow \infty$, then $B^o \rightarrow B^*$.*

Lemma 1 indicates that as long as the total sensing period T is of sufficient length, the diminution in budget consumption optimality caused by the loss of complete future information is negligible. Hence, both B^o and B^* can serve as the same benchmark for an online policy that does not require future information. An online policy is necessary as the stochastic future information required by (21) may not be available in practice. ARA is thus modelled as an online problem of reward allocation, i.e., with no future information. The offline problem serves as a benchmark only.

4.6. Online Budget Consumption Optimization Problem. The Lyapunov Optimization-based budget optimization problem

formulated in this section relies only on past response rates to particular rewards and does not require any future information. The goal of the service provider is to minimize the time average reward and hence optimize its budget consumption. The service provider's budget (B) consumed in time slot t is given by

$$B(t) = R(t) N(t) \quad (22)$$

Lyapunov Optimization requires a control decision. For ARA, the control decision refers to the setting of an optimal reward level $R(t)$ for a particular time slot t . Thus, $R(t)$ is the control decision made in time slot t . The resultant reward allocation policy arising from $R(t)$ must meet constraints (14), (15), (16), and (17).

The time average budget consumption of this policy can then be defined as

$$B_{AV} \triangleq \lim_{t \rightarrow \infty} \frac{1}{t} \sum_{\tau=0}^{t-1} E \{ R(\tau) N(\tau) \} \quad (23)$$

The goal of ARA's reward model is to determine a reward level $R(t)$ that minimizes the time average budget consumption subject to constraints (14), (15), (16), and (17).

4.7. Designing the Reward Algorithm. The virtual queue, $Z_{\text{forfeit}}(t)$, in the modelled system is the dimension that has to be considered to achieve an optimal reward for a time slot t . As a result, from (9), the Lyapunov function for t can then be defined as

$$L(t) \triangleq \frac{1}{2} Z_{\text{forfeit}}(t)^2 \quad (24)$$

Equation (24) is a quadratic Lyapunov function, a scalar measure of the total queue backlog in the participatory sensing system. The expected change in the Lyapunov function over one time slot t is referred to as the one-slot conditional Lyapunov drift and is defined as

$$\Delta(t) \triangleq L(t+1) - L(t) \quad (25)$$

To achieve adaptive reward allocation that minimizes the reward offered for a data submission (and thus optimizes budget consumption) and still obtain meaningful and timely responses for the service provider's dataset, (25) must be greedily minimized for each timeslot t (i.e., the solution that is the best for the current timeslot is chosen) so as to minimize the queue backlog. In queuing theory terms, this means that the queue backlogs are pushed towards a lower congestion state on an ongoing basis with the goal of achieving queue stability. Therefore the budget consumption term $B(t)$ is incorporated into (25) to produce a *drift-plus-penalty* expression:

$$\Delta(t) + V \{ B(t) \} \quad (26)$$

Given that the overall objective is to minimize budget consumption, it should be minimized at the same time as the queue backlog is being minimized. This minimization

objective is known as a *penalty* under Lyapunov Optimization. The fundamental objective of Lyapunov Optimization is to minimize the bound (limit) on the drift-plus-penalty expression [35]. V is a nonnegative control parameter that is used to incorporate the weighted budget consumption term in the control decision. This facilitates the trade-off required by the service provider between reducing the backlog of Z_{forfeit} and minimizing B . Thus, in statistical terms, the goal is to find the upper bound for (26).

The drift-plus-penalty bound for a general case [35] can be extended for the environment in which ARA operates. The number of responses received for an offer, $N_{\text{received}}(t)$, is assumed to be i.i.d. over time slots. Therefore, under any control algorithm that seeks to minimize the reward allocated, $R(t)$, the drift-plus-penalty expression used for Lyapunov Optimization [35] can be formulated for ARA with the following upper bound:

$$\begin{aligned} \Delta(t) + V \{B(t)\} \\ \leq B_{\text{constant}} + VE \{B(t) \mid Z(t)\} \\ + Z(t) E \{N_{\text{received}}(t-1) - N_{\text{desired}}(t) \mid Z(t)\} \end{aligned} \quad (27)$$

It should be noted that B_{constant} is a positive number used in the Lyapunov Optimization computation and is defined by

$$B_{\text{constant}} \triangleq \frac{1}{2} (N_{\text{received}}(t-1) - N_{\text{desired}}(t))^2 \quad (28)$$

Like other Lyapunov Optimization-based models [35], the objective of the reward allocation algorithm presented for ARA is not to directly minimize (26). The goal rather is to minimize the upper bound on the right hand side of (27). Therefore, the reward allocation algorithm observes the queue backlog $Z(t)$ in every time slot t and adapts the Lyapunov Optimization approach [35] to choose the budget consumption $B(t)$ as the solution to the following problem:

$$\min N(t) (VR(t) + Z(t)) \quad (29)$$

As was noted in (1), $N(t)$ is a function of $R(t)$. This constraint is ensured by the supply curves and thus the solution to problem (29) must be one of the rewards depicted on the relevant curve for the current time slot. This means that the reward to be allocated, $R(t)$, can only be one of a number of possible values for each time slot t . The algorithm evaluates (29) for all possible levels of budget consumption and selects the reward corresponding to the optimal level of consumption. After this reward is selected, the responses are processed and rewarded by the service provider. The appropriate supply curve is then updated to reflect $N(t)$, the number of responses received. The execution of the algorithm is repeated for every time slot in which an offer is made.

A typical Lyapunov Optimization model only requires the current system state. This is modified for ARA as the algorithm determines the reward $R(t)$ to offer on the basis of the number of responses received in previous timeslots. In other words, the algorithm offers higher rewards when the backlog for Z_{forfeit} is large and lowers the level of reward to offer when the backlog for Z_{forfeit} is small.

The optimality of (29) can be proven using standard Lyapunov Optimization theory [35]. $B^\dagger(t)$ denotes the budget consumption in a timeslot t . Using B^* , the budget consumption benchmark that assumes stochastic future information, the following theorem can be presented.

Theorem 2 (adapted from [35]).

$$\lim_{t \rightarrow \infty} \sum_{t \in T} E(B^\dagger(t)) \geq B^* - \frac{B_{\text{constant}}}{\phi} \quad (30)$$

Equation (30) implies that the formulation for the online budget consumption optimization converges to the minimum budget consumption asymptotically (as time tends towards infinity), with a controllable error bound $O(1/\phi)$.

4.8. Incorporating Data Utility. The value of V is a key factor in devising an optimal budget consumption policy [33]. Specifically, if B_{av}^* is the objective value of the time average maximization problem under an optimal policy, the following theorem holds [36].

Theorem 3 (adapted from [36]). Suppose the number of responses received $N_{\text{received}}(t-1)$ and the number of desired responses $N_{\text{desired}}(t)$ are i.i.d. for each time slot. If there exists an $\delta > 0$ such that

$$E\{N_{\text{received}}(t-1)\} \leq E\{N_{\text{desired}}(t)\} - \delta \quad (31)$$

the following performance guarantees are then realized:

$$\begin{aligned} \lim_{t \rightarrow \infty} \frac{1}{t} \sum_{t=0}^{t-1} E\{p(t)N(t)\} &< B_{\text{av}}^* + \frac{B_{\text{constant}}}{V} \\ \bar{Z} \triangleq \lim_{t \rightarrow \infty} \frac{1}{t} \sum_{t=0}^{t-1} E\{Z(t)\} \\ &\leq B_{\text{constant}} + \frac{B_{\text{constant}} + VR_{\text{max}}P_{\text{max}}}{\epsilon} + N_{\text{max}} \end{aligned} \quad (32)$$

$p(t)$ is the penalty used for achieving queue stability in Lyapunov Optimization (budget consumption in this case) while ϵ represents a constant > 0 .

Theorem 3 indicates that, by choosing a large value for V , the budget consumption can be arbitrarily close to the optimal solution. However, the average queue backlogs increase as the value of V is increased. This means that there is a trade-off between budget consumption and the size of $Z(t)$ that can be tuned by the service provider depending on the significance of the data it is seeking in a particular timeslot, t .

As the importance of data being sought will vary for the service provider, it can set a utility weighting U for these data submissions. The utility weighting increases with the importance of the data to the service provider and can be used to capture dynamic changes in the participating sensing environment. To reflect the importance of the data being sought, the value of U is mapped to that of V . Specifically, the value of V is increased in accordance with the data utility


```

// Construct a Linear Regression model from the supply curve.
Create Supply Curve for {[Nactual], R} from Historical Dataset
Prediction_Model = Linear Regression Model for
                    Supply Curve {[Nactual], R}
// Predict the responses for different reward levels.
[R] = 0..Rmax
[Npredict, R] = predict(Prediction_Model, [R])

// Construct the queuing state variables for each reward.
foreach R
    Zforfeit = Npredict - Ndesired
    {[R, Npredict, Zforfeit(t)]} +=
    [R, Npredict, Zforfeit(t)]
end foreach

// Compute the constant used for Lyapunov Optimization.
Bconstant = 0.5 * (Nreceived(t-1) - Ndesired(t))2

// Map the data utility weighting U to the value of V
V = [U, V]
// Evaluate each reward using Lyapunov Optimization.
foreach [R, Npredict, Zforfeit(t)]
    // Compute the Budget used by this reward.
    B = R * Npredict
    // Check that the budget consumption
    // does not exceed the set maximum.
    if B > Bproportion_max then
        break
    // Carry out the Lyapunov Optimization
    // computation.
    L = 1/2 * Zforfeit(t)2
    // Compute the one slot conditional Lyapunov drift.
    Δ(t) = L - Llast(R)
    // Evaluate the drift plus penalty expression.
    DPPLHS = Δ(t) + (V * B)
    DPPRHS = Bconstant + (V * B)
    + Z(t)(Nreceived(t-1) - Ndesired(t))
    if DPPLHS >= DPPRHS then
        continue
    end if
    // Evaluate the current optimization
    // computation.
    OPTcurrent = Npredict * ((V * R) + Z(t))
    if OPTcurrent > OPTsolution then
        OPTsolution = OPTcurrent
        Roptimal = R
    end if
end foreach

```

ALGORITHM 1: Reward computation.

weighting so as to prioritize attracting data submissions over budget consumption, i.e.,

$$U \propto \frac{1}{V} \quad (33)$$

Utility weighting can thus be used to capture dynamic changes in the participating sensing environment. It should be noted that the predictive model could also be used to tune

the utility of the sought data submission without the need to modify the ARA algorithm. For example, the most recent data received could be weighted when constructing the predictive model if data is being sought on the basis of the most recent submissions.

Algorithm 1 presents the algorithm for reward computation. Table 1 presents the additional notations used in this algorithm.

TABLE 1: Notations.

DPP_{LSH}	Left hand side of Drift Plus Penalty expression.
DPP_{RSH}	Right hand side of Drift Plus Penalty expression.
$L_{last}(R)$	Last Lyapunov function calculated for a particular reward.
O_{total}	Total Number of Offers
$OPT_{current}$	Current Lyapunov Optimization calculation.
$OPT_{solution}$	Lyapunov Optimization solution.
$[P]_{count}$	Number of Participants.
predict	Predict using Linear Regression model.
$R_{optimal}$	The optimal value for the reward.
$[R]$	Set of possible reward values.
$R(t)$	The optimal reward to offer in a particular timeslot, t .
$[R, N_{predict}, Z_{forfeit}(t), Z_{lower}(t)]$	Reward, number of predicted responses and queuing state variables for this reward.
$\{[R, N_{predict}, Z_{forfeit}(t)]\}$	The set of rewards, their respective queuing state variables and number of predicted responses.
$\{[N_{actual}], R\}$	Actual responses for the different reward levels.
$[N_{predict}, R]$	The number of predicted responses for the different reward levels.
N_{total}	Total number of responses
$[U, V]$	A map of data utility weightings and the constant V used for computing the Lyapunov Drift.
ΔX	One-slot conditional Lyapunov Drift.

5. Evaluation of ARA

ARA is evaluated through a series of experiments carried out in a simulated participatory sensing environment. The algorithm has been implemented for the simulation using the C++ and Statistical R programming languages.

Section 5.1 presents the experimental setup for this environment. Section 5.2 evaluates the adaptiveness of ARA with respect to the response rate and the utility of the data sought while Section 5.3 assesses the budget consumption of the approach. The SenseUtil approach [17] and the STOC-PISCES algorithm [25] are used as the baselines for comparison as the objective of the former is to compute incentives on the basis of data utility while the goal of the latter in adapting rewards to the response rate and utility is similar to that of ARA.

5.1. Experimental Setup. ARA is evaluated in a simulated participatory sensing environment. In this simulation, the service provider makes a series of offers over the duration of the simulation. 100 responses are sought for each offer. While the number of responses sought will vary among participatory sensing applications as well as over time, this

TABLE 2: Simulation parameters.

Simulation Duration	3600 seconds
Offer Interval	30 seconds
No. of Responding Participants	10-200
Initial Reward	105
No. of Responses Sought	100
Initial Response Rate	
(i) High Response Environment	70%-200%
(ii) Low Response Environment	10%-50%
Maximum Reward for an Offer	200
Service Provider Budget	50,000
V	0,1000
Rolling Window for Regression	100

figure is chosen to clearly determine whether the reward level is adapting to the response rate and the utility of the data sought. The maximum reward to be set for an offer is 200 units. Each simulation runs for one hour with offers being generated every 30 seconds. For the purposes of the simulation, it is assumed that each offer corresponds to one timeslot t (i.e., one offer is produced per timeslot) with the reward being reevaluated with each offer.

The simulation is run in two types of environment, one with a high initial number of responses (referred to as the “high response environment”) and one with a low initial number of responses low (referred to as the “low response environment”). The initial response rate ranges between 70% and 200% and 10% and 50%, respectively, for these environments. The participant response rate is generated using a continuous uniform distribution. The simulation model varies this response rate using a randomly generated increment to evaluate how the reward adapts to these changes in the response rate. This randomness is incorporated to reflect other factors in the participatory sensing environment that may affect the response rate.

For the purposes of the simulation, the response rate is calculated simply as the ratio of the number of responses submitted to the number of responses sought. However, the response rate could be defined using other metrics such as the coverage of a particular area ([37, 38]) without impacting the underlying algorithm.

As noted in Section 4.8, the value of V can be tuned to reflect the data attributes that are of most interest to a service provider at a particular point in time. For the purposes of the evaluation, the value of V for each offer is set either to 0 to prioritize attracting data submissions or 1000 to prioritize budget consumption. As V is set to 0 for the majority of experiments, the value of V is only indicated when its value is 1000.

Finally, the size of the rolling window used for predicting the number of responses is set to the last 100 responses while the budget is set to 50,000 units for those experiments that evaluate budget consumption. Table 2 presents the parameters used for the simulation.

While the similar objectives of ARA and STOC-PISCES mean that the latter can be integrated in the modelled

FIGURE 1: Adapting the reward to the response rate.

participatory sensing environment without customization of the underlying algorithm, this is not the case for SenseUtil as the latter does not adapt rewards to the response rate. Rather, SenseUtil determines the reward to offer on the basis of the number of potential participants. To ensure a valid comparison, the SenseUtil model is simulated in a participatory sensing environment with the number of potential participants set to 50 (the figure used by the authors) and 200 (which, according to the authors of the approach, should lead to lower rewards), respectively. The computed utility, as is the case for the simulation used by the authors, is mapped on a one-to-one basis to an economic point system which in turn determines the reward to be offered. A one-to-one mapping between this economic point system and the reward to be offered is used for the simulation with the utility range being set from 10 to 200 to correspond to the reward range used for ARA and STOC-PISCES. Distance between sensed locations is used by SenseUtil to determine data utility with the minimum distance used to compute the location utility in the simulation being (like the authors' simulation) set to 50 m and 100 m.

5.2. Adaptiveness and Utility. Figure 1 presents the adaptiveness of ARA to the response rate. It can be seen from the graph that the reward is increased so as to attract more data submissions at low response rates while the reward is reduced where the response rate approaches or exceeds 100%. Moreover, the reward settles on a value over time that generates a response rate close to 100%. It should be noted that the reward is not always immediately adapted after a change in the response rate for a particular offer as the regression model used for the supply curves ensures that the focus is on changes that occur over time rather than sudden changes that may be outliers, thus ensuring that the budget is not needlessly consumed.

Figures 2–6 compare the adaptiveness of ARA with the STOC-PISCES algorithm. The number of initial trials used by STOC-PISCES is set to 10, which is the figure used by the approach's authors in their evaluation. To ensure a fair comparison, the initial reward for ARA is set to an initial value of 105 units. This is because the STOC-PISCES

FIGURE 2: ARA adaptiveness (high response environment).

FIGURE 3: ARA adaptiveness (low response environment).

FIGURE 4: ARA adaptiveness (low response environment, $V = 1000$).

algorithm initially runs a number of trials offering an initially higher reward at the median (105 for a range of 10–200).

It can be seen from Figures 2 and 5 that STOC-PISCES adapts to the response rate at a much slower rate than ARA in a high response environment. This is reflected in the average reward offered by STOC-PISCES which, at 102.87 units, is substantially higher than the figure of 12.53 units for IPPI.

The findings for a low response environment for ARA and STOC-PISCES are presented in Figures 3 and 6, respectively.

FIGURE 5: STOC-PISCES adaptiveness (high response environment).

FIGURE 6: STOC-PISCES adaptiveness (low response environment).

In this environment, STOC-PISCES rapidly and substantially increases the reward it offers so as to attract more submissions. This leads to much higher rewards being offered for the equivalent response rate received by ARA. ARA offers a substantially higher average reward of 48.26 units in this environment which is, nonetheless, still much lower than the average of 175.25 units offered by STOC-PISCES.

Figure 4 shows that the average reward is significantly lower (11.08 units) when, by setting the value of V to 1000, budget consumption is prioritized over attracting data submissions in a low response environment. This highlights how ARA not only adapts its reward to response rates but also uses the value of V to take data utility into account.

5.3. Budget Consumption. Figure 7 presents the average reward for ARA, STOC-PISCES, and SenseUtil. It can be seen that the average reward for ARA is lower than that computed by the other two approaches. This is the case even in a low response environment with V being set to zero so as to attract as many responses as possible.

Figures 8 and 9 present a comparison of the budget consumption by ARA and the STOC-PISCES algorithm. The

FIGURE 7: Average reward.

SenseUtil approach is not used for this experiment as integrating adaptiveness to the response rate would have required modification and extension of the underlying algorithm. Each simulation has been run until the allocated budget has been consumed.

It can be seen from Figure 8(a) that, with a budget of 50,000 units, ARA generates 3325 responses in a high response environment. This figure is significantly larger than that for STOC-PISCES at 598. Moreover, in a low response environment, ARA generates over 3682 responses, albeit with a much higher number of offers than in the high response environment. It should also be noted that, in both the low and high response environment, the budget optimization of ARA is superior to that of STOC-PISCES as the former generates a higher number of offers (as shown in Figure 8(b), the number of offers is 31 and 102, respectively, for ARA; the number of offers is 6 and 13, respectively, for STOC-PISCES) with the same budget.

The higher number of responses generated by ARA appears to result from STOC-PISCES not taking budget consumption into account. Specifically, the offering of the same reward for a number of trials regardless of the response rate results in a higher overall average reward and more rapid budget consumption. In contrast, ARA reduces the reward it offers more quickly in a high response environment and while it does increase its reward in a low response environment, it does so more prudently than STOC-PISCES which tends to raise the level of reward offered close to the maximum reward more quickly than ARA. This is borne out by the budget consumption which is much steadier for ARA in both a low and a high response environment as shown in Figure 9.

6. Conclusion

This paper proposes ARA, an adaptive incentivization scheme that uses the Lyapunov Optimization method to provide rewards that seek to capture data that reflects the dynamic environment in which many participatory sensing applications operate. Experimental results show that ARA performs ongoing reward computation on the basis of response rates and data utility.

This work addresses the fundamental challenge of rewarding participants in a way that ensures that the service

FIGURE 8: Responses and offers.

FIGURE 9: Budget consumption.

provider's dataset is reflective of the participatory sensing environment. Further challenges to be addressed include the need to make ARA privacy aware, secure, and incentive compatible so as to provide untraceable rewards and preserve identity privacy as well as evaluating the approach's robustness to collusion attacks that artificially raise reward prices.

Data Availability

The programming code used to support and evaluate ARA is available from the authors upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

The authors acknowledge the support of their employers, Cork Institute of Technology and Trinity College Dublin.

References

- [1] R. Szabo, K. Farkas, M. Ispany et al., "Framework for smart city applications based on participatory sensing," in *Proceedings of the 4th IEEE International Conference on Cognitive Infocommunications (CogInfoCom '13)*, pp. 295–300, Budapest, Hungary, December 2013.
- [2] B. Predic, Z. Yan, J. Eberle, D. Stojanovic, and K. Aberer, "ExposureSense: Integrating daily activities with air quality using mobile participatory sensing," in *Proceedings of the 2013 IEEE International Conference on Pervasive Computing and Communications Workshops, PerCom Workshops 2013*, pp. 303–305, USA, March 2013.
- [3] A. Clarke and R. Steele, "Targeted and anonymized smart-phone-based public health interventions in a participatory sensing system," in *Proceedings of the 2014 36th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, EMBC 2014*, pp. 3678–3682, USA, August 2014.
- [4] D. Christin, C. Büchner, and N. Leibecke, "What's the value of your privacy? Exploring factors that influence privacy-sensitive contributions to participatory sensing applications," in *Proceedings of the 2013 IEEE 38th Conference on Local*

- Computer Networks Workshops, LCN Workshops 2013*, pp. 918–923, Australia, October 2013.
- [5] J. Mohite, Y. Karale, P. Gupta, S. Kulkarni, B. Jagyasi, and A. Zape, “RuPS: rural participatory sensing with rewarding mechanisms for crop monitoring,” in *Proceedings of the 13th IEEE International Conference on Pervasive Computing and Communication (PerCom Workshops '15)*, pp. 378–383, St. Louis, Mo, USA, March 2015.
 - [6] A. Mao, E. Kamar, Y. Chen et al., “Volunteering versus work for pay: Incentives and tradeoffs,” in *Proceedings of the First AAAI Conference on Human Computation and Crowdsourcing*, 2013.
 - [7] Y. Wang, A. Nakao, and A. V. Vasilakos, “Heterogeneity playing key role,” *ACM Transactions on Autonomous and Adaptive Systems (TAAS)*, vol. 7, no. 3, pp. 1–25, 2012.
 - [8] L. Gao, F. Hou, and J. Huang, “Providing long-term participation incentive in participatory sensing,” in *Proceedings of the 34th IEEE Annual Conference on Computer Communications and Networks (IEEE INFOCOM '15)*, pp. 2803–2811, Hong Kong, China, May 2015.
 - [9] J. Ren, Y. Zhang, K. Zhang, and X. Shen, “SACRM: Social Aware Crowdsourcing with Reputation Management in mobile sensing,” *Computer Communications*, vol. 65, pp. 55–65, 2015.
 - [10] W. Wang, H. Gao, C. H. Liu, and K. K. Leung, “Credible and energy-aware participant selection with limited task budget for mobile crowd sensing,” *Ad Hoc Networks*, vol. 43, pp. 56–70, 2016.
 - [11] X. Sun, J. Li, W. Zheng, and H. Liu, “Towards a sustainable incentive mechanism for participatory sensing,” in *Proceedings of the 1st IEEE International Conference on Internet-of-Things Design and Implementation, IoTDI 2016*, pp. 49–60, Germany, April 2016.
 - [12] B. O. Holzbauer, B. K. Szymanski, and E. Bulut, “Socially-aware market mechanism for participatory sensing,” in *Proceedings of the the first ACM international workshop*, pp. 9–14, Istanbul, Turkey, August 2012.
 - [13] L. G. Jaimes, A. Chakeri, J. Lopez, and A. Raij, “A cooperative incentive mechanism for recurrent crowd sensing,” in *Proceedings of the SoutheastCon 2015*, pp. 1–5, Fort Lauderdale, FL, USA, April 2015.
 - [14] X. Zhang, Z. Yang, Z. Zhou, H. Cai, L. Chen, and X. Li, “Free market of crowdsourcing: Incentive mechanism design for mobile sensing,” *IEEE Transactions on Parallel and Distributed Systems*, vol. 25, no. 12, pp. 3190–3200, 2014.
 - [15] T. Sandholm, “Issues in computational vickrey auctions,” *International Journal of Electronic Commerce*, vol. 4, no. 3, pp. 107–129, 2000.
 - [16] M. Kumar and S. I. Feldman, “Internet Auctions,” in *Proceedings of the USENIX Workshop on Electronic Commerce*, vol. 3, pp. 49–60, August 1998.
 - [17] T. Tsujimori, N. Thepvilajanapong, Y. Ohta, Y. Zhao, and Y. Tobe, “History-based Incentive for Crowd Sensing,” in *Proceedings of the the 2014 International Workshop*, pp. 1–6, Saint Etienne, France, September 2014.
 - [18] J. Zhang, J. Ma, W. Wang, and Y. Liu, “A novel privacy protection scheme for participatory sensing with incentives,” in *Proceedings of the 2012 IEEE 2nd International Conference on Cloud Computing and Intelligence Systems (CCIS)*, pp. 1017–1021, Hangzhou, China, October 2012.
 - [19] A. Singla and A. Krause, “Incentives for Privacy Tradeoff in Community Sensing,” in *Proceedings of the 1st AAAI Conference on Human Computation and Crowdsourcing (HCOMP-13)*, 2013.
 - [20] G. Radanovic and B. Faltings, “Incentive schemes for participatory sensing,” in *Proceedings of the 14th International Conference on Autonomous Agents and Multiagent Systems, AAMAS 2015*, pp. 1081–1089, Turkey, May 2015.
 - [21] Z. Song, E. Ngai, J. Ma, and W. Wang, “A novel incentive negotiation mechanism for participatory sensing under budget constraints,” in *Proceedings of the 22nd IEEE International Symposium of Quality of Service, IWQoS 2014*, pp. 326–331, Hong Kong, May 2014.
 - [22] C. H. Liu, P. Hui, J. W. Branch, C. Bisdikian, and B. Yang, “Efficient network management for context-aware participatory sensing,” in *Proceedings of the 2011 8th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks, SECON 2011*, pp. 116–124, USA, June 2011.
 - [23] X. Niu, M. Li, Q. Chen, Q. Cao, and H. Wang, “EPPI: An E-cent-based privacy-preserving incentive mechanism for participatory sensing systems,” in *Proceedings of the 33rd IEEE International Performance Computing and Communications Conference, IPCCC 2014*, pp. 1–8, USA, December 2014.
 - [24] S. Yang, U. Adeel, and J. McCann, “Backpressure meets taxes: Faithful data collection in stochastic mobile phone sensing systems,” in *Proceedings of the 34th IEEE Annual Conference on Computer Communications and Networks, IEEE INFOCOM 2015*, pp. 1490–1498, Hong Kong, May 2015.
 - [25] A. Biswas, D. Chander, K. Dasgupta et al., “PISCES: participatory incentive strategies for effective community engagement in smart cities,” in *Proceedings of the Third AAAI Conference on Human Computation and Crowdsourcing*, 2015.
 - [26] C. Huang, L. Sankar, and A. D. Sarwate, “Designing Incentive Schemes for Privacy-Sensitive Users,” *Journal of Privacy and Confidentiality*, vol. 7, no. 1, 2016.
 - [27] Y. Liu, C. Yuen, N. Ul Hassan, S. Huang, R. Yu, and S. Xie, “Electricity Cost Minimization for a Microgrid with Distributed Energy Resource under Different Information Availability,” *IEEE Transactions on Industrial Electronics*, vol. 62, no. 4, pp. 2571–2583, 2015.
 - [28] R. Urgaonkar, U. C. Kozat, K. Igarashi, and M. J. Neely, “Dynamic resource allocation and power management in virtualized data centers,” in *Proceedings of the 12th IEEE/IFIP Network Operations and Management Symposium (NOMS '10)*, pp. 479–486, April 2010.
 - [29] C.-Y. Lee and H. Heo, “Estimating willingness to pay for renewable energy in South Korea using the contingent valuation method,” *Energy Policy*, vol. 94, pp. 150–156, 2016.
 - [30] C.-K. Tham and T. Luo, “Quality of contributed service and market equilibrium for participatory sensing,” *IEEE Transactions on Mobile Computing*, vol. 14, no. 4, pp. 829–842, 2015.
 - [31] S. Kalish and P. Nelson, “A comparison of ranking, rating and reservation price measurement in conjoint analysis,” *Marketing Letters*, vol. 2, no. 4, pp. 327–335, 1991.
 - [32] R. Carter Hill, *Principles of Econometrics*, Wiley, 2011.
 - [33] X. Labandeira, J. M. Labeaga, and X. López-Otero, “A meta-analysis on the price elasticity of energy demand,” *Energy Policy*, vol. 102, pp. 549–568, 2017.
 - [34] A. San Martini and F. Spezzaferrri, “A predictive model selection criterion,” *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 46, no. 2, pp. 296–303, 1984.
 - [35] M. J. Neely, “Stochastic network optimization with application to communication and queueing systems,” *Morgan Claypool (Synthesis Lectures on Communication Networks)*, vol. 7, 2010.

- [36] M. J. Neely and R. Urgaonkar, "Opportunism, backpressure, and stochastic optimization with the wireless broadcast advantage," in *Proceedings of the 2008 42nd Asilomar Conference on Signals, Systems and Computers, ASILOMAR 2008*, pp. 2152–2158, USA, October 2008.
- [37] M. Girolami, S. Chessa, M. Dragone, M. Bouroche, and V. Cahill, "Using spatial interpolation in the design of a coverage metric for Mobile CrowdSensing systems," in *Proceedings of the 2016 IEEE Symposium on Computers and Communication, ISCC 2016*, pp. 147–152, Italy, July 2016.
- [38] M. Girolami, S. Chessa, G. Adami, M. Dragone, and L. Foschini, "Sensing Interpolation Strategies for a Mobile Crowdsensing Platform," in *Proceedings of the 5th IEEE International Conference on Mobile Cloud Computing, Services and Engineering, MobileCloud 2017*, pp. 102–108, USA, April 2017.